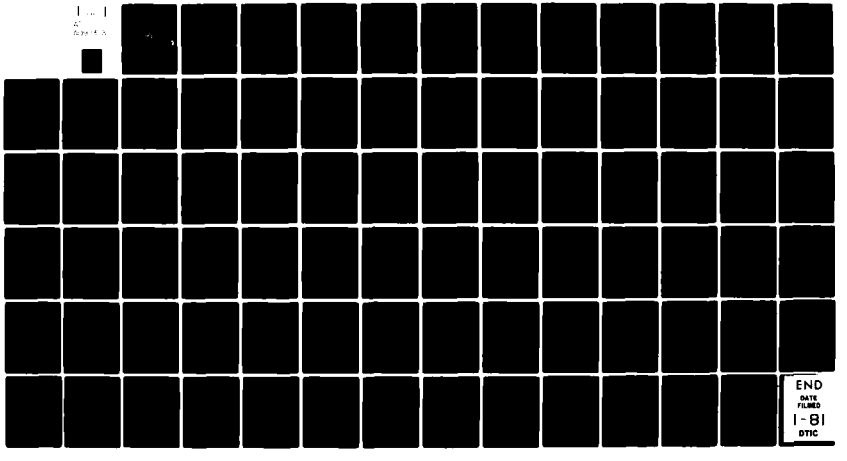
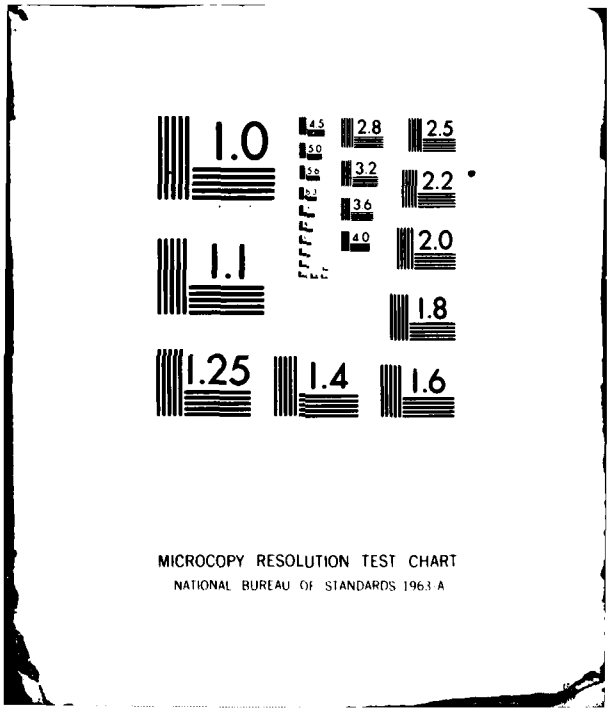
AD-A092 363 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH F/G 13/8
A NEW COMPUTER PROGRAM FOR PLANT LAYOUT DESIGN - OPDEP OPTIMAL --ETC(U)
1980 A C NELSON
UNCLASSIFIED AFIT-CI-80-16T NL



END
DATE FILMED
1-81
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

AFIT-CI-80-16T

UNCLASS
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: 80-16T	2. GOVT ACCESSION NO.: AD-A092363	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): A New Computer Program for Plant Layout Design - OPDEP Optimal Plant Design and Evaluation Program		5. TYPE OF REPORT & PERIOD COVERED: THESIS/DISSERTATION
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Allan C. Nelson		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: AFIT STUDENT AT: Iowa State University		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 12 81
11. CONTROLLING OFFICE NAME AND ADDRESS: AFIT/NR, WPAFB OH 45433		12. REPORT DATE: 1980
		13. NUMBER OF PAGES: 73
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report): UNCLASS
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

LEVEL II

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DTIC ELECTE DEC 3 1980 D

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

C

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: IAW AFR 190-17

FREDRIC C. LYNCH, Major, USAF
Director of Public Affairs

Air Force Institute of Technology (ATC)
Wright-Patterson AFB, OH 45433

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ATTACHED

JCB

80 11 24 069

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASS
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD A092363

DTIC FILE COPY

ABSTRACT

Statement of Objective

The objective of this paper is to present a detailed explanation of the OPDEP plant layout program, and contrast it with ALDEP, the program upon which it is based. An explanation of the ALDEP program is presented first, along with a critique of its performance, then two improved versions are discussed. The first improved version, ~~is~~ called ALRANDOM, ~~which~~ merely generates layouts according to the modified output of a random number generator. The second version, ~~is~~ called OPDEP, This program is considerably more sophisticated than the others, and generates its layouts according to a complex heuristic algorithm. This algorithm optimizes the department selection process by analyzing the portion of the layout already completed before making the next department selection. Detailed instructions for using the program, as well as methods for getting the best results from the program are provided. Finally, a comparison of the three programs on some example problems and conclusions on their performance ~~is~~ are presented.

80-16T

A new computer program for plant layout design - OPDEP Optimal Plant Design and Evaluation Program

by

Allan C. Nelson

A Thesis Submitted to the Graduate Faculty in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

Major: Industrial Engineering

Approved:

Howard H. Smith
In Charge of Major Work

Keith R. McBride
For the Major Department

C B Tanas
For the Graduate College

Iowa State University
Ames, Iowa

1980

80-16T

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to ascertain the value and/or contribution of research accomplished by students or faculty of the Air Force Institute of Technology (ATC). It would be greatly appreciated if you would complete the following questionnaire and return it to:

AFIT/NR
Wright-Patterson AFB OH 45433

Research Title: A New Computer Program for Plant Layout Design - OPDEP Optimal Plant Design and Evaluation Program

Author: Allan C. Nelson

Research Assessment Questions:

1. Did this research contribute to a current Air Force project?

a. Yes b. No

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not?

a. Yes b. No

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency achieved/received by virtue of AFIT performing the research. Can you estimate what this research would have cost if it had been accomplished under contract or if it had been done in-house in terms of manpower and/or dollars?

a. Man-years _____. b. $_____

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3 above), what is your estimate of its significance?

a. Highly Significant b. Significant c. Slightly Significant d. Of No Significance

5. AFIT welcomes any further comments you may have on the above questions, or any additional details concerning the current application, future potential, or other value of this research. Please use the back of this questionnaire for your statement(s).

NAME GRADE POSITION

ORGANIZATION LOCATION

USAF SCN 75-208

TABLE OF CONTENTS

CHAPTER I. INTRODUCTION

Review of Previous Work

One of the earliest works involving a systematized technique of plant layout can be traced to "Systematic Layout Planning", by Richard Muther (1). He attempts to provide procedures with sufficient structure to permit practical problems to be solved economically with a systematic approach. His book introduced the relationship chart and the relationship classifications which later became the basis for a number of the computer plant layout routines.

Computer programs for facilities layout have been in use since 1963, when CRAFT (Computerized Relative Allocation of Facilities Technique) was introduced (2). Computerized plant layout routines may be classified as either 1) construction type or 2) improvement type. The construction algorithms build or construct a solution in an open floor area from the raw data. The improvement algorithms require a feasible solution as part of the input. The program works on this feasible solution and improves it until no further improvements can be found.

A recent survey (1974) conducted by J.M. Moore (3) indicated that there were approximately twice as many construction routines as improvement routines. The following

routines were identified as construction routines, the asterisk indicates the program is widely used:

ALDEP *	GENOPT	LSP
COLO2	IMAGE	MUSTLAP2
COMP2	KONUVER	PLAN
COMSBUL	LAYADAPT	PLANET *
CORELAP *	LAYOPT	SISTLAP
DOMINO	LAYOUT	SUMI

The following routines were identified as layout improvement routines:

ALDEP *	GRASP	OFFICE
COFAD *	KONUVER	PREP
COSFAD	LAYADAPT	SET
CRAFT *		

According to a 1975 survey conducted by the Facilities Planning and Design Division of AIIE (American Institute of Industrial Engineers), approximately one in three of the participants of the survey have had some experience with computer programs designed to assist in layout work. Eighty-three percent of those experienced with computer aided layout programs responded that the programs were being used either to generate alternative layouts or to evaluate alternative layouts. The survey also indicated some dissatisfaction with the inability of most plant layout programs to generate practical or usable layouts, but most users indicated the programs were of some value in the early stages of design work in eliminating many of the inefficient or unuseable layouts.

The author, in preparing to write the OPDEP program, obtained copies of several of the above programs and information on most of them by writing to the addresses listed in J. M. Moore's previously referenced international survey.

COFAD is one of the most recent programs developed, and also appears to be one of the most useful (4). It is a sophisticated version of CRAFT which includes expanded cost and alternative material handling data that is used in designing the layout. The layouts generated by this program are quite realistic but seem to lack imagination and creativity.

CORELAP is another routine which seems to give generally good results on most layout problems (5). It is unique in its design approach in that the first department placed in the layout is always the one which has the highest cumulative preference for other departments. The first department is always placed in the middle of the layout to ensure adequate space for all preferred departments to be placed around it. Of all the layout algorithms investigated by the author, CORELAP seems to generate the most consistently good designs. The logic that ensures the first department to place will always be one with a high number of preferences for other departments, also prevents the program (CORELAP)from experimenting with other layouts using different departments at the center. This tends to limit the flexibility and

creativeness of the designs. It seemed to the author that the primary advantage of the computer is its speed in creating and evaluating different designs. Therefore, it did not seem logical to unnecessarily limit the creative potential of the design program. This was the approach taken in the development of OPDEP.

Many of the routines are restricted to very specialized portions of plant layout problems and cannot be used for actual design or evaluation of designs (IMAGE, CRAFT, LAYOPT) (6),(7). Others can be used for design but only to the degree that the building must be built around the layout (KONUVER, PLANET, SISTLAP) (3),(8). The most versatile of the programs studied by the author was ALDEP (9), because so many restrictions on the design can be imposed by the user. Unfortunately the ALDEP program, though versatile, does not always produce results that are very useful to the plant layout designer.

It would be a very interesting experiment to compare all of these programs on a standard layout problem and make a relative ranking of their effectiveness based on a comparison of their designs. However, the evaluation of the designs would have to be standardized and conducted on the basis of the same criteria to make the ranking a valid one. The output from each of these programs is quite different and to evaluate them all by the same method would be very difficult. The author decided

on the basis of a subjective comparison that the program with the most potential for pure design work was ALDEP, even though its designs were not as good as several of the other programs.

The author was first introduced to the ALDEP program while using it to make plant layouts for an Industrial Engineering Design course at Iowa State University. It was discovered, after running the program extensively, that the layouts produced were of little value and the program seemed very insensitive to changes in the input data. To make this program more useful as a plant layout design tool, it seemed desirable to make some improvements in the way the program interpreted and used the data input to it. Therefore, the author undertook the task of rewriting the program. As it turned out, the most difficult thing to do was to fully understand the original program. This was partly due to the lack of comments used by the original programmer, as well as to the early unstructured style of Fortran programming used in writing the program.

ABSTRACT

Statement of Objective

The objective of this paper is to present a detailed explanation of the OPDEP plant layout program, and contrast it with ALDEP, the program upon which it is based. An explanation of the ALDEP program is presented first, along with a critique of its performance, then two improved versions are discussed. The first improved version is called ALRANDOM, which merely generates layouts according to the modified output of a random number generator. The second version is called OPDEP. This program is considerably more sophisticated than the others, and generates its layouts according to a complex heuristic algorithm. This algorithm optimizes the department selection process by analyzing the portion of the layout already completed before making the next department selection. Detailed instructions for using the program, as well as methods for getting the best results from the program are provided. Finally, a comparison of the three programs on some example problems and conclusions on their performance is presented.

CHAPTER II. THE LOGIC OF THE PROGRAMS

Explanation of ALDEP

The ALDEP program was written by Wayne O. Evans of IBM in 1967. The program is marketed by IBM and has been used widely as a computer aid to plant layout. The program consists of a main program and 4 subroutines; they are: DECODE, ASSIGN, LAYOUT and EVALU. The main program first initializes the various arrays and variables used throughout the program and assigns the required constants. Next, the input data, with the exception of preassignment data, are read in. The input data will be explained in detail in Chapter III - Example Problem, so it is only listed here. The input data consist of a date and run code, random number generator seed, design parameters to control the patterns used in the layout process, a variable format for each of the three floors, square size, rounding factor, number of layouts desired, minimum acceptable score to print out, value of a necessary relationship, department numbers and areas, preference matrix and finally preassignments, followed by a trailer card to indicate the end of the run or the start of another data set.

The DECODE subroutine is then called to convert the letters A, E, I, O, U, and X in the preference matrix to the built-in integer values of the program. The preference matrix

is a matrix whose row and column intersections contain the desired relationship between the department represented by the row, and the department represented by the column. The A preference represents a relationship between two departments which Absolutely requires they be close to each other. The E preference represents a relationship between two departments which Especially requires they be close to each other. The I preference represents a relationship between two departments signifying it is Important they be close to each other. The O preference represents a relationship in which closeness is Optional. The U represents a relationship in which closeness is Unnecessary, and the X represents a relationship in which closeness is eXtraordinarily undesirable (1). The built in values corresponding to the letters A through X are 64, 16, 4, 1, 0, and -1028 respectively. Obviously A represents the most important positive relationship and the other letters represent a less important positive relation in descending order to X which has a negative relationship. The DECODE subroutine translates this matrix of letters into one containing only integers for more efficient interpretation by the program.

The ASSIGN subroutine is called next to read in data on departments that are restricted to a particular floor in a multifloor layout, or to restrict departments to specific squares within a floor. This prevents these departments from

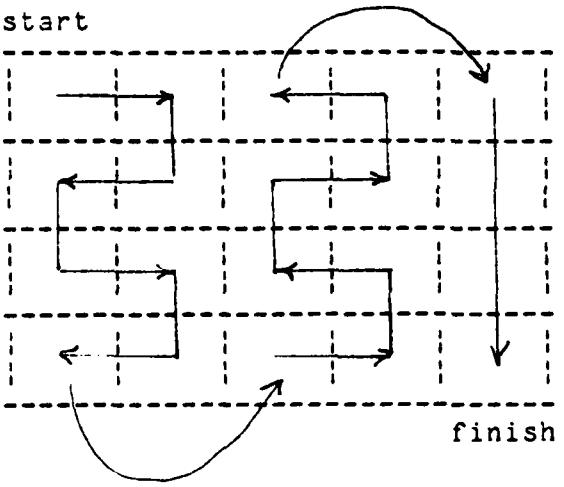
being laid out at the discretion of the program. To do this, the ASSIGN subroutine tabulates a matrix of these preassignments, and then equates it to the layout matrix at the start of every layout, so preassigned departments will always be in the same location. ASSIGN also builds a two dimensional array called KFLOR, which contains the department numbers to be laid out on each floor and keeps track of the departments which have and have not been laid out as the design progresses.

The LAYOUT subroutine is called repeatedly for as many layouts as are specified, to perform the actual design process. The first step is to generate a randomly ordered array of numbers which correspond to the numbers of the departments being laid out (no duplicate numbers). LAYOUT uses a self-contained random number generator, identical to the RANDU subroutine available with most Fortran compilers. In ALDEP, the random array NRANDM is used to select the first department to lay out each time it is called and also to choose subsequent departments, when one cannot be found by searching the preference matrix. The first number in NRANDM, which was not preassigned is used as the first department to lay out. The first number which has not been laid out or preassigned is used when choosing the next department to lay out.

The next step in LAYOUT is to verify that the departments

with their respective areas, as input, can be fitted into the dimensions of the building specified in the input data. If not, an appropriate diagnostic message is printed out; otherwise the program continues and begins the actual design work.

The dimensions of the layout matrix and the input sweep length are combined to make an if loop/do loop combination, which determines the pattern the layout design will follow. The design starts in the upper left hand corner of the layout matrix and moves right in a do loop until the sweep length is reached as the upper limit of the do loop. Then the row number is incremented by one, which moves the pattern down to the second row. Here the if loop takes over and moves from right to left until it has moved left a distance equal to the sweep length. The row number is incremented again to move the pattern down to the third row and the do loop begins again to move right. This zig zag pattern continues until the bottom of the matrix is reached. At this point, the upper and lower limits of both loops are adjusted so the pattern continues just to the right and moves upward. The same sequence is continued until the matrix has been completely filled, or there are no more departments to place. The following diagram shows this pattern graphically:



sweep length= 2

building width= 5

building depth= 4

start in column 1

This design pattern can be modified by changing the sweep length parameter or the column to start placement parameter. Now that the pattern is determined, the manner in which departments are selected for placement will be discussed. As mentioned previously, the first department to place is always the first number contained in the NRANDM array that has not already been preassigned. This department is laid out in the design pattern until the number of squares it is to occupy is reached. The number of squares is contained in the variable KNT which is reset each time a department is selected. The program first tests the particular square it is about to fill with the current department number, to make sure it is empty (contains a zero). If it is not empty the design pattern is continued until the next empty square is found. This prevents overwriting on top of a square which was intended to be

preassigned. Preassigned means the user has specified that a department be placed in a specific square or squares.

The program is now ready to lay out the next department. To move the program in the right direction in its search for good layouts, the author of ALDEP used a search procedure on the decoded preference matrix to select subsequent departments to lay out. This search procedure takes the number of the department just laid out as the column number of the preference matrix to search. However, it does not search all the positive values for a maximum. It only searches until it finds a value for a preference greater than or equal to the input value of a necessary relationship, hereafter referred to as MUST. This, in effect, tells the LAYOUT subroutine that a preference must be at least equal to this value to be considered for selection as the next department to place, provided it has not already been laid out or is not preassigned. The problem with this approach is that to have all preferences considered in the design as they should be, the value of MUST has to be specified as some relatively small number, say 4 or 1. Then, when LAYOUT does its search, it may find a 4 as one of the first preferences in the column and will select that department number which corresponds to the preference as the next one to place. The search stops there. However, there may very well be several other values in the column that are much higher and would have been better

choices.

Evans (9), the author of ALDEP, attempts to compensate for this problem by recommending that only the A preference value be used as the value of the necessary relationship. This only prevents the program from considering preferences that are smaller than the maximum value, forcing the use of the random number array as the primary means of selecting a department, when all the values in the preference table should be used. What about the case of a column that contains several preferences smaller than MUST? The ALDEP program does not consider these and goes on to select the next department randomly with little chance of it being the one which will optimize the layout at that point. The search procedure makes no provisions for the case where several preferences that are greater than or equal to MUST are found, but have the same value (ties). The procedure always chooses the first one found that has not been placed. This logic steers the program in the same direction on each design, while random selection among the tied departments may steer the program towards some altogether different and preferable layout.

Evans was attempting to guide the program in the selection process, while letting the variability that is inherent in a random number generator, eventually lead to an optimal design. What actually occurs in most layout problems is that the search procedure consistently guides the layout in

the same direction, so the same designs are found over and over again, and they are by no means the best ones.

The following table presents a very simple preference matrix which will illustrate the nonoptimality of this search procedure.

Table 1 Example Preference Matrix *

	1	2	3	4	5	6	7	8
101	0	0	32	0	32	32	64	1
102	0	0	4	32	0	32	0	1
103	32	4	0	0	1	64	64	16
104	0	32	0	0	16	0	64	16
105	32	0	1	16	0	16	0	0
106	32	32	64	0	16	0	16	0
107	64	0	64	64	0	16	0	64
108	1	1	16	0	0	0	64	0

* Assume for this example that:
NRANDM = 108,101,102,107,106,
103,105,104 and MUST = 64

In this example, 108 would be the first department placed by the program since it occurs first in the NRANDM array. The column corresponding to 108 is searched first to find that department whose preference to be placed next to 108 is the highest. The ALDEP procedure operates satisfactorily on this column because there is a value equal to must in the seventh row which corresponds to department 107. Therefore, department

107 will be placed next, since it has not been placed already and is not preassigned. The column for 107 is now searched; the procedure fails here because both 103 and 104 have a preference of 64, but since 101's preference occurs first, they are not even considered and have no chance to be selected on subsequent runs beginning with 108. Instead, 101 is selected and laid out. When 101's column is searched, the search fails since the only 64 belongs to 107, which has already been laid out. Departments 103, 105 and 106 would have been good choices but were ignored by the search. The program is forced to use the random number array for selection and 102 is chosen since 103 and 101 have been laid out. Department 102 is obviously not the best choice. The sequence can be continued to demonstrate other nonoptimal choices resulting from this search procedure. If the ALDEP search procedure fails frequently enough the program chooses the departments randomly and eventually designs some good layouts. More often than not however, the sequence becomes locked into some undesirable order, and the good designs are never found. The LAYOUT subroutine continues in this manner until all the departments have been laid out, then the EVALU subroutine is called to score the design just completed by LAYOUT.

EVALU scores each square in the layout matrix individually, according to the preference matrix. At the start of the evaluation, a copy of the preference matrix is made

from the original and used one time only. Each preference that is satisfied by the layout is added to the layout score and then set to zero in the copy, until all the squares have been evaluated. At the end of the evaluation, each preference that has not been satisfied is still in this copy of the preference matrix, but all the others are zero. To find those preferences not satisfied by the layout, the program tallys all those that remain in the copy of the preference matrix. This list is printed below each layout matrix on the program output. The value of a necessary relationship, that was input by the user, determines the minimum preference to consider in making the list of preferences that were not satisfied. If a value of 64 (A) is input as the value of a close relationship, then only the A preferences that were not satisfied are listed. In a large complex layout problem, it is not unusual to have several pages of unsatisfied preferences listed, if a small value is input for the value of a close relationship. EVALU considers a preference to be satisfied when the two departments involved are laid out adjacent to each other, either diagonally or perpendicularly. If one square separates two departments, their preference is not scored as satisfied. This is a feature of the evaluation subroutine that can cause good layouts to be overlooked by the program. A large number of departments can be laid out with one square separating them, so that their preference is not added to the score in

one layout. In another layout, these same departments can be laid out separated by many squares and receive the same score, provided that everything else is the same. In other words, the "all or none" criterion of the evaluation considers a short separation to be as bad as a large separation between departments.

Explanation of ALRANDOM

The ALRANDOM program is very much like ALDEP, except that the preference matrix is ignored entirely during the design process, and is used only to evaluate the design after it is completed. The selection sequence of subsequent departments is done entirely by the random number generator, so that any tendency towards a certain selection sequence is theoretically eliminated. If there is a bias in the random number generator, then of course this bias will affect the selection sequence also. The serious disadvantage of this program is that although it is free to pursue any possible sequence of department selections, the number of different random number arrays possible for an average layout problem is astronomical. For a small number of departments, say 20, the number of different randomly ordered arrays containing the numbers 1 through 20 is 20 factorial (!) or 2.432902 times 10 to the

18th. That is 2.43 billion billion different possibilities. Experience with the program indicates that for an average layout of 20 departments, approximately 5 layouts can be designed, evaluated and tested / second of CPU time on the Digital - VAX 11/780 system. Therefore, a minimum of 15,429,363,000 years of CPU time on the VAX system would be required to design, evaluate and test all the possible layouts from the ALRANDOM program. This is assuming that the random number generator is a perfect one and that its period of repetition is larger than 20!. Most computerized random number generators have a period of repetition that is a large number, but one might wonder if the period could be large enough to give satisfactory results on a problem such as this one.

To determine if the period of the random number generator is a limiting factor in the performance of the ALRANDOM program, the author wrote a program to find the average number of cycles the generator goes through before the first duplicate of the first NRANDM array is generated. Theoretically, the average number of cycles a perfect generator should go through before such a duplicate array is generated is N!, where N is the number of elements in the array. This is because these various numbers of cycles required by the generator conform to the geometric distribution. The expected value of the mean of the geometric distribution is given by 1/P, where P is the probability of

generating a duplicate in any given attempt (10). For example with an array size of 6 randomly ordered numbers between 1 and 6 (each number is contained only once in an array) there are 6! or 720 different arrays that can be generated. Therefore, over a large number of generations, the average cycles between duplication of the first array should tend toward this number of different possible arrays (720). This expectation is confirmed by the program, at least for the small array sizes (3,4,5,6,7,8,9 and 10) that are practical, in terms of cpu time for this test program to process. This test program is contained in Appendix B. It should be reasonable to extrapolate these results to the larger array sizes, to the extent that the duplications of any given array will not occur in the 9,999 layouts that are possible in any one run of this program. Therefore, the random number generator used in these programs (UNI) should not be a limiting factor. In all the array sizes mentioned above, the average number of cycles of the generator required before the first duplicate (of the object array) was found, was within 1 percent of the factorial of the array size. Layouts with more than 8 departmen' s (8! is 40,320) require such a large amount of computer time for ALRANDOM to design all possible layouts, that it becomes impractical to use. Unfortunately, the larger the layout is, the more benefit a layout optimization program can be in designing the layout.

As fast as the latest generation of digital computers has become, they are not yet fast enough to make this random design approach practical. Some form of optimization must be used to eliminate the vast majority of unacceptable layout designs.

This program was written to serve as a baseline with which to compare the other programs. In order to tell how effective the program is in developing good layouts, it is necessary to know what part of the layout score can be attributed to random chance. This can be determined by comparing the score of ALRANDOM on a particular data set to the other programs using the identical data set. ALRANDOM uses the same random number generator, so meaningful comparisons can be made among the three programs, provided the same generator seeds are used. If significant improvement in the layout scores results from using ALDEP rather than ALRANDOM on identical data, then it can be concluded that the program interaction with the selection sequence has been effective. This same type of test can be made among any of the three programs for comparison purposes. The results of these tests are presented in Chapter IV of this paper.

The Development of OPDEP

The author's first attempt to improve ALDEP involved replacement of the section of code in the LAYOUT subroutine which searches the preference matrix with a call to another subroutine called SELECT. SELECT, in this first version, was a relatively short subroutine of about 80 statements which searched the appropriate column of the preference matrix. The search procedure used was considerably different than the one used in ALDEP; no minimum value was specified to use in searching the column in the preference matrix. The search was performed for all positive preference values. These values were placed in an array and the maximum value or values were found and placed in another array. If ties existed in this array, a random method of selecting one of the tied departments was employed. The random number array NRANDM was used to settle the tie situation by finding which of the tied department numbers occurred first in the NRANDM array, and then selecting that department to be placed next by the LAYOUT subroutine. As an example of this tie breaking method, assume the column search has discovered three departments: 3, 7, and 8 which all have a preference of 64, to be placed next to the department just placed by the program. Assume further that the random number array is: 4, 1, 7, 3, 5, 6, 2, 8. Since 7 occurs

first in the array, it will be selected as the next department to place. This technique provides enough variability to insure that when several equal choices are available, eventually all will be tried and the design will not be locked into some narrowly defined sequence of selections. This first version was called ALDEP2.

ALDEP2 resulted in significant improvement in every layout problem attempted. Good designs were generated much sooner than with ALDEP, and very good ones were eventually generated that could never be generated using ALDEP. There was still a problem, however, when preassigned departments were placed in the layout matrix.

When the search of the preference matrix is done, the previously placed department number is used as the column number to search in the preference matrix. To receive consideration as a department to place some other department next to, the department must first be placed by the LAYOUT subroutine. Then the department number is passed to SELECT as the one for which to optimize the next selection. Preassigned departments are never placed by the LAYOUT subroutine. Because they are in the layout matrix at the time LAYOUT is called, they receive no more consideration than the boundaries of the matrix.

Another shortcoming of ALDEP2 was its failure to fully utilize available information on the departments already in

the program, those remaining to be placed, and their respective preferences. It was decided that LAYOUT should pass the coordinates of the next empty square in the design pattern to SELECT rather than passing the last department number placed. Then, SELECT should look at all the squares which surround the one passed to it and see which of these 8 squares already have departments contained in them. The ones which contain numbers should all be given consideration when attempting to optimize the choice of the next department to place. In this manner, departments in any of these 8 squares which have been placed by the program, or preassigned by the user, would receive equal consideration by the program.

To accomplish this increased "awareness" by the program required only a few changes to the LAYOUT subroutine, but SELECT had to be increased in length and complexity by a considerable degree. SELECT is now over 200 statements long and has a number of nested do loops, so the number of statements executed each time SELECT is called has increased greatly.

SELECT now begins by checking what is contained in each of the 3 squares surrounding the square whose coordinates have been passed to it. The nonzero numbers found in these squares are checked for duplicates, and placed in an array called PREFR without the duplicates. These values in PREFR represent departments which are immediately adjacent to the square about

to be laid out; and their respective columns in the preference matrix are all searched in precisely the same manner as done in ALDEP2. Ties are also settled in the same manner as before.

Each time a column is searched and a maximum selected, its value is placed in WINPRF. After all the appropriate columns have been selected and their selected preference placed in WINPRF, this array of winning selections is searched for a "maximum of the maximums". If no ties are found for the maximum, the department number which corresponds to the maximum is selected to be placed next. If ties occur in this WINPRF array, the following procedure is followed: The department numbers that correspond to the preferences in WINPRF are placed in WINTIE and it is checked to see if it contains duplicate department numbers. If it contains duplicates, the duplicate number which occurs most frequently is selected as the final choice and returned to LAYOUT. The reason for this is that duplicate department numbers at this stage indicate that placing the duplicate department will satisfy at least two separate high preferences from at least two departments surrounding the square to be laid out. If no duplicates are found in WINTIE, then the ties that are in it for maximum preference are settled in the usual random manner.

This final selection is passed back to LAYOUT as the next department to place in the design. Rarely, it occurs that no preference can be found by SELECT (usually when there are only

a few departments left to place). In this case, the LAYOUT subroutine is signaled by SELECT to make its own choice (random selection).

A basic assumption has been made by the author in developing this program. The assumption is that the objective of the design process is to place the departments which have material handling activity between them as close to each other as possible, with priority given to higher activity rates (11). This is the same basic assumption stated by Muther (1) in his previously mentioned book on plant layout. The author acknowledges that this may be an oversimplification in certain situations, however it is generally a valid assumption that the cost of material handling is directly proportional to the distance moved and the number of trips made (12).

EVALU has also been modified from the original version used in ALDEP which only checked the 8 squares immediately surrounding each square it evaluated in a design. The new version of EVALU checks these 8 surrounding squares through the entire design and records the score. Then it checks the 16 squares in the group immediately outside the first group of 8 to see what preferences are satisfied with one square between the two departments. To illustrate this concept:

9	10	11	12	13
24	1	2	3	14
23	8	X	4	15
22	7	6	5	16
21	20	19	18	17

X represents the square being evaluated
1 - 8 represent the inner square group,
9 - 24 represent the outer group.

The inner group of squares is scored as before but the outer group is scored at one half the preference value when a department in the outer group has a preference for the square being evaluated. When the second scoring sequence is completed, those preferences half satisfied are tallied and the additional score is added to the first sequence score. At the bottom of each layout, all preferences not satisfied are printed out along with their letter value. At this point, a note is also printed out stating how many preferences were half satisfied and the amount they contributed to the layout score. This method gives the evaluation a broader view of the whole layout and allows more flexibility in the best designs generated. It may seem that a preference could be scored twice this way, or that a higher than perfect score could be achieved, but this is not the case.

The main program has been greatly expanded over the original one to provide more information about the layout in the program output. Significant data parameters are printed

out to enable the user to more easily find mistakes in the input data. The program now summarizes, for all 3 floors, the number of departments to place, the area required for placement, the area available for placement, the preassigned areas and the buffer areas in the layout. This summary is printed in the program output to allow the user to verify if all input data has been interpreted as intended and to make errors in problem set up easier to locate. The addition of an identifier for each workcenter, printed at the start of the output as well as in a legend below each layout, makes the layouts more readable and less symbolic in nature. The main program also does the input and output file handling and provides prompts and column numbers for all the required input as an aid in entering the data. Error checking has also been expanded by providing diagnostic messages for any input conversion problems encountered by the program. The theoretical perfect score of the job is now calculated by the program, as well as the total number of preferences to satisfy, and how many are satisfied by any layout that is printed out. Statistical analysis of the layout scores is also provided at the end of the output; the highest scoring design, the lowest scoring design and the average of all designs.

The DECODE subroutine has been changed to allow the user to specify any 2 digit number as the value of the preferences A - X instead of the constant values required by ALDEP. If no

preference values are supplied to the program, it provides default values.

This new version is called OPDEP (for Optimal Plant Design and Evaluation Program), because it tends more toward optimal layouts than ALDEP does. The OPDEP program contains approximately 1400 executable statements and requires 256K of memory compared to 500 executable statements and 128K of memory for ALDEP. The OPDEP program is written in Digital's "Fortran IV - Plus" language, but can easily be made compatible with other extended Fortran languages such as WATFIV or Fortran 77. The program is intended for use with a CRT terminal system rather than card input but can be adapted to batch input on cards if desired.

General Information on the OPDEP Program

A brief explanation of the way OPDEP handles the input/output files and an overview of the program operation is now provided. The OPDEP user is initially confronted with several questions to answer concerning the files to be used for data input or data saving, and whether the output will come to the terminal directly or be saved in a file for later print-out. The file opening and closing is done by the program; the user need only specify the file name to be used.

There are two restrictions on file names. First, when the program asks what file the data is in, the user must enter a valid file name. Second, a file name must be less than 10 letters long with a period after the name followed by three letters, e.g. FILENAME1.DAT.

The user has the option of supplying data to the program directly via prompted input or by specifying an existing file as the location for the program to get the required data. If the first option is chosen then the user will be asked for a file name in which to store the data as it is entered for subsequent reuse. If a file name is not supplied, the program points out that the data will be lost, then continues. If the second option is chosen then a data file must already exist. A data file will exist once data has been entered with prompts because the program will save the data in a specified file for reuse later. This saves considerable time for the user in not reentering the entire file each time the program is run, because the program is, in effect, creating a "data card deck". Next the program asks where to send the output. The choices are either directly to the terminal or to a specified output file. The file must be less than 10 letters long with a period and any three letters after the period. If there is something wrong with a file name specified, the program will print an error message and terminate execution. Prompts are supplied by the program, telling the user what to enter and

what columns to enter it in. After the prompt message, all column numbers are labeled to make entering the data less confusing for the user. Some error checking is done by the program as the data is read in with appropriate error messages printed out when required.

The program continues to produce layouts until 9,999 per run or all those requested have been designed. Only those that score above some minimum score supplied by the user are printed on the screen or saved in a file. If the output is sent to a file rather than the screen, the user can follow the progress of the program by a statement sent to the screen telling the score of the current layout, if it was saved and how many have been saved thus far.

The preference matrix is the most critical part of the data set used by OPDEP. The program designs its layouts and evaluates them by closely following the preference matrix. A high scoring layout or "perfect layout" will only be as good as the input preferences, therefore considerable work should go into making this preference matrix as accurate and practical a representation of inter-departmental relationships as possible. A preference matrix not carefully analyzed and prepared according to some consistent criteria will result in plant layouts that may score high, but in reality will be of little value in developing a final plant layout.

The matrix may be constructed based on any criteria

chosen, but the best results are generally obtained by using material handling activity between departments as the primary criterion for establishing preferences between departments. In this program, an A represents the highest amount of material handling activity, with E,I and O representing progressively lesser amounts of activity. U represents no material handling activity and X represents some undesirable relationship such as the president's office being located next to a 500 ton press. It should be noted here that the X preference should be used sparingly since it tends to result in low scoring layouts if used excessively with a negative value associated with it. If used with a zero or positive value as the lowest level of material handling activity, then the X preference does not reduce the quality of the layouts.

The numeric values of these preferences are supplied by the user and have no specific units associated with them, but the intended units are pallets/day or parts/day or barrels/week, etc. In other words, some consistently applied unit size that is associated with material handling activity must be used. If, for instance, the highest material handling activity in a layout problem is 25 pallets/day between departments then the user may wish to choose the numeric value of A as 25. If the next highest amount of material handling activity is say 15 pallets/day then the value of E may be chosen as 15 and so on down to a value of 0 for U. When

choosing these values one should keep in mind that with only 5 relative steps to work with (A,E,I,O,U) not every different material handling activity value that occurs in the layout problem can be represented by a different preference letter. So one may have to group material handling activity values and assign some average value to the preference letters. Inclusion of the X preference as one that represents material handling activity gives 6 relative steps.

If it is desired to have two departments laid out close to each other for some reason other than material handling activity, then a high preference such as A or E may be entered in the matrix to insure their proximity in the layouts. This should only be done when necessary because these "artificial preferences" tend to complicate the layout if used excessively.

The program disregards the size of differences between preferences, discerning only that one is larger or smaller than another. In searching the preference matrix the program looks for the highest preference in a particular column (whether the highest number is 1000 greater than all the others or only 1 greater does not matter in choosing the highest preference). Its choice will always be the highest value and in case of ties a choice is made among the tied departments randomly to allow for variability in layout design. The effect of large differences in magnitude between

preference values is that it makes the layout scores biased towards those which satisfy the relationships with the large preference values. Normally this is a desirable thing since the user will want the higher material handling activity departments close to each other. An example layout problem, presented in the next few pages will illustrate these concepts and serve as a model for setting up data files for other plant layout problems.

CHAPTER III. HOW TO USE THE OPDEP PROGRAM

Example Layout Problem

Consider a hypothetical factory which produces parts for lawnmowers. It has 16 depts and the layout is restricted to an area of 5000 sq ft approximately 63 ft by 79 ft.

The first line of data required by the program is the date and the name of the layout. The date must be entered in cols 2-21 and the layout name must be entered in cols 22-41. Here is the way the first line appears:

December 1, 1979 RECOIL STARTER MFG.

The next line contains two seeds for the random number generator. These numbers must be odd, positive integers 4 digits long and they are entered in cols 2-5 and 6-9, as:

13573579

The next lines contain data which control the layout size and design. Three lines of data are entered here, one for each

possible floor. The first line pertains to the top floor, the second to the main floor, and the third to the basement. If any of the floors are nonexistent in the layout, a blank line is left for that floor. In our hypothetical case, only one floor is available so the first and third of these lines will be left blank (the main floor is used in one floor layouts). All three lines have the same format when used. Cols 2-3 contain the col number to begin placement. Always use 01 here. Cols 4-5 contain the sweep length to be used in designing the layout (the number of squares to move right or left before going to the next row). Small numbers (02,03,04) generally work best. Cols 6-9 contain the number of squares available to layout on this floor. Before doing this one must decide on the square size to use in the layout. One approach is to use the size in square feet of the smallest dept as the square size for the layout. In this case it is 250.0 square ft. Since the square root of 250 is about 15.8 one can fit 20 squares into the floor space of 5000 sq ft or 5 squares wide by 4 squares deep. So in cols 6-9 0020 is used. In cols 10-13 enter the width of the floor in squares (0005), in cols 14-17 enter the depth of the floor in squares (0004). The next three lines look like this:

```
0000000000000000
0102002000050004
0000000000000000
```

The next data line contains the square size in sq ft, the rounding factor to use in calculating the number of squares for each department, the number of layouts to run, the minimum score required to be saved and the minimum preference to consider in listing those not satisfied by a particular layout. In cols 2-6, enter the number of square feet in a square with a decimal point (250.0). In cols 7-10, enter the rounding factor (use 00.6 always). In 11-14, enter the number of layouts to be run, maximum of 9,999 (say 0005 for this example), note that it takes a long time to run more than 500 layouts. In cols 15-18 enter the minimum layout score required to be saved (0650 is used here but for a first run the user will just have to guess or use 0). In cols 19-21, enter the minimum preference value to be considered in printing out those preferences not satisfied for a given layout (001 is used here but on some layouts one may wish to use a larger number to limit the list length). The next line looks like this:

250.000.600050650001

The next data line contains 3 fields which specify the format to be used for the top, main and basement floors. If no

layout is to be done on a particular floor enter (02I3) in that respective field. This is because a border of zeros is printed around each floor and if there is no layout to make, then only two rows of two zeros are printed out to signify an empty floor. In cols 2-7, enter the format for the top floor layout (since there is no top floor (02I3) is entered). In cols 11-15, enter the main floor format (since there is a main floor enter an integer 2 larger than the width of the layout in squares, this layout is 5 squares wide so use (07I3)here). In cols 18-23, enter the format for the basement; since there is no basement (02I3) is used. The next data line looks like this:

(02I3) (07I3) (02I3)

The next series of lines are department data lines. They contain a department number starting with 101 and progressively increasing in a continuous manner through the last department. Each line also contains a size specification in square feet with a decimal pt., and an identifier for the department up to 20 characters long. At the end of these department data lines a blank line is inserted to signal the end of this type of data. In cols 2-4 enter the 3 digit department number beginning with 101 up to a maximum of 163,

for 63 departments. In cols 7-13, enter the department area in square feet with a decimal point. In cols 16-36 enter the name of the department. This series of lines for the example looks like this:

```
101   500.       RECEIVING RAW MAT.
102   250.       RECEIVING PCH. PTS.
103   500.       SHEAR
104   250.       PUNCH PRESS
105   250.       PLASTIC MOLDING
106   250.       BRAKE PRESS
107   250.       FORM PRESS 250T
108   250.       ARC WELD
109   250.       WELD CLEANING
110   250.       SUBASSEMBLY
111   250.       PAINTING
112   250.       DEBURRING
113   250.       DRILL PRESS
114   250.       WIP FOR ASSEMBLY
115   500.       ASSEMBLY
116   500.       SHIPPING/WHSE
```

The next sequence of lines contains the preference matrix. Suppose that from the from-to charts and other material handling data for the hypothetical factory, the material handling flow between workcenters has been summarized in the units of forklift trips per week as follows: 101 to 103 is 24 trips/wk, 101 to 105 is 7 trips/wk; 102 to 103 is 7 trips/wk, 102 to 110 is 3 trips/wk; 103 to 104 is 25 trips/wk, 103 to 106 is 6 trips/wk, 103 to 107 is 20 trips/wk;104 to 106 is 26 trips/wk, 104 to 107 is 3 trips/wk,

104 to 108 is 7 trips/wk; 105 to 112 is 6 trips/wk; 106 to 108 is 26 trips/wk; 107 to 110 is 19 trips/wk, 107 to 112 is 7 trips/ week; 108 to 109 is 25 trips/wk; 109 to 110 is 25 trips/wk; 110 to 111 is 25 trips/wk; 111 to 113 is 8 trips/wk, 111 to 114 is 20 trips/wk; 112 to 115 is 6 trips/wk; 113 to 114 is 19 trips/wk; 114 to 115 is 21 trips/wk; and 115 to 116 is 28 trips/ wk. It is possible to categorize these material handling intensities into A for 28 trips/wk, E for 24 trips per week, I for 20 trips per week, O for 7 trips per week, U for 0 trips per week and an X relationship is not used in this problem. Enter in cols 2-4 the department number in the same order as the department data lines above. In cols 5-67 enter the preferences A,E,I,O,U,and X with S for the diagonal. Use capital letters only. Assembling all this material handling data into a matrix results in a preference matrix which looks like this:

```
101S
102US
103EUS
104UUAS
105OUUS
106UUOAUS
107UUIOUUS
108UOUOUAUS
109UUUUUUUES
110UOUUUUIUES
111UUUUUUUUUES
112UUUUOUOUUUUS
113UUUUUUUUUUUOUS
114UUUUUUUUUUUIUIS
```

115UIUUUUUUUUUOUIS
116UUUUUUUUUUUUUUAS

The next line contains the preference integer values. In cols 3-4 enter the value of A (28). In cols 9-10 enter the value of E (24). in cols 15-16 enter the value of I (20). In cols 21-21 enter the value of O (7). In cols 27-28 enter the value of U (0), and in cols 33-34 enter the value of X (0). These values must be right justified in the field. This next line looks like this:

28 24 20 7 0 0

The next lines contain preassigned area information for the program to build the preassigned matrix. Suppose that the truck dock for the factory is located at the northwest corner of the building and that the user does not wish to relocate it, so the receiving raw materials department must remain at its present location. OPDEP allows for this type of restriction. Since dept 101 has 500 square feet it will require 2 squares to be preassigned for it. The coordinates of these two squares, because they are located at the far left of the North wall will be 1,1 and 2,1. Suppose the user wants to keep the Shipping/ warehouse (department 116) in its present location at the far right of the North wall in two squares

whose coordinates will be 4,1 and 5,1. The column numbers in the example go from 1 on the far left to 5 on the far right, and the row numbers go from 1 at the top to 4 at the bottom of the layout matrix. If the user wished to preassign the bottom right square in the matrix, its coordinates would be 5,4. The next few lines give complete instructions for entering preassignment information.

If no preassigned areas are desired, enter a capital A in col 2 and hit RETURN. If a particular dept is to be restricted to a certain floor in a multifloor layout, enter an F in col 2, the dept number in cols 6-8, and the number of the floor (1 is top, 2 is main and 3 is basement) in col 10. If a department is desired in a particular square or squares of the layout matrix, enter an A in col 2, a 1 in col 4 (use 2,3,4 etc. if the line is the second, third or fourth continuation line of a department since only 14 squares can be designated on one line). Enter the department number in cols 6-8, enter the floor number in col 10 and enter the number of squares the department will occupy in cols 11-13(right justify). Enter the coordinates of each square the department will occupy starting with the col# in cols 14-15 (right justify), the row# in cols 16-17(right justify), and so on for each square to be occupied by the department. Use 2 cols for the col# and 2 cols for the row#. Up to 14 squares can be designated on one line, then it becomes necessary to use continuation lines. When all

preassigned areas have been entered, on the next line put an A in col 2 and hit RETURN. The preassignment lines for the example look like this:

```
A  1 101 2  2 1 1 2 1
A  1 116 2  2 4 1 5 1
A
```

The last line of data required in the data file is the answer to a question from the program. The question is: Do you want the sequence of scores generated by the program printed out? The only time this option is of any value to the user is when the run is made with new data and the minimum score is set too high, so no layouts are saved. The score printout option allows one to see what range the scores were in, so the minimum score can be readjusted downward. The user could also look at the average score statistic at the end of the output. As a general rule the answer to the question should be NO or blank spaces entered in cols 2-4. This option was added to the program as a convenient means of obtaining all the layout scores for statistical normality testing. The complete data file of the example problem is provided here:

```
December 1, 1979       RECOIL STARTER MFG
13573579
000000000000000000
0102002000050004
000000000000000000
250.000.600050650001
(02I3)  (07I3)  (02I3)
101   500.      RECEIVING RAW MAT.
102   250.      RECEIVING PURCH.
103   500.      SHEAR
104   250.      PUNCH PRESS
105   250.      PLASTIC MOLDING
106   250.      BRAKE PRESS
107   250.      FORM PRESS-250T
108   250.      ARC WELD
109   250.      WELD CLEAN
110   250.      SUBASSEMBLY
111   250.      PAINTING
112   250.      DEBURRING
113   250.      DRILL PRESS
114   250.      WIP ASSEMBLY
115   500.      ASSEMBLY
116   500.      SHIPPING/WHSE

101S
102US
103EUS
104UUAS
105OUUUS
106UUOAUS
107UUIOUUS
108UOUOUAUS
109UUUUUUUES
110UOUUUUUIUES
111UUUUUUUUUUES
112UUUUOUOUUUUS
113UUUUUUUUUUOUS
114UUUUUUUUUUIUIS
115UIUUUUUUUUUOUIS
116UUUUUUUUUUUUUUAS
 23     24     20     7       0       0
A 1 101 2  2 1 1 2 1
A 1 116 2  2 4 1 5 1
A
NO
```

Using this file, scores in the 90 to 95% range are easily achievable within the first 100 layout attempts. The

percentages refer to percent of a perfect score, where the perfect score is calculated by adding up all the preferences in the preference matrix, then multiplying by 2. The 2 factor comes about because each preference entered is considered to be 2 preferences by the program. For example if 101 has a preference of 24 for 103, then the program also counts this as 103 having a preference for 101 of 24. In the example problem the theoretical perfect score is 795. The original ALDEP program from IBM, modified to use the same VAX random number generator, the same scoring subroutine and the same data file, is unable to design a layout that scores above 84% in 5000 attempts. This data file is available for use as test data in the OPDEP.TST file. The user may wish to run the program with this file to become familiar with the program output.

Some Suggestions for Better Results

The program will generally produce layouts in the 75% and higher range for any data set, unless the preference matrix is unusually complex and large. The user should be cautioned against placing unnecessary constraints on the layout by making the preference matrix with extraneous relationships between departments. really not necessary. If no necessary relationship exists between two departments, then enter it as

a U relationship and not some other letter. Too many preferences in the matrix take some flexibility away from the designs and unnecessarily complicate the design process. Another common mistake is to make the square size too small. There is no advantage to a square size smaller than the smallest department area.

A technique that proves helpful in finding the highest possible score for a layout involves running the program the first time for about 200 layouts. Then, if some of the highest scoring layouts have departments laid out in an acceptable are, preassign the departments to that area for the next run. Run the program for another 200 layouts, increasing the minimum score slightly to keep the volume of output down. Repeat the same procedure on a few more departments until a satisfactory layout is obtained. This procedure tends to guide the program in the right direction and eliminate some of the lower scoring layouts. Also, since the program looks at preassigned areas as well as areas already laid out when selecting the next department to place, the preassigned areas won't lower the score, as they do in the IBM version.

Another method of finding higher scoring designs is to vary the sweep length parameter. Values from 01 through the width of the building may be tried for this parameter. Do not use a value larger than the width of the building in squares or the layout matrix will be distorted. This technique will

usually uncover some better layouts than can be found using just one value of the sweep parameter.

On complex layouts it generally is not possible to achieve a perfect score. However, scores in the 75 to 85% range are usually achievable within a few hundred attempts.

In evaluating any layout produced by the program certain factors should be considered. Some of these factors are:

1. The possible lack of conformity of department shapes.
2. The possibility of unreasonable results such as the shipping department being placed in the middle of the building.
3. The lack of concern for overall facility flexibility and expandability.
4. The need for aisle space and work in process area.

It is hoped that the user will enjoy letting the computer "do the work" once the data file has been written. The program is of the most benefit on those layouts that are large and complex. It does a good job on smaller ones also, but if they are very small, they may be laid out just as well by hand. This program, if properly used, should save time for the user in finding some quite good layouts which can almost always be improved upon by careful analysis.

CHAPTER IV. RESULTS AND CONCLUSIONS

Comparison of Program Results

Throughout the development of OPDEP, the program's layouts were compared with those of ALDEP and ALRANDOM on a number of layout problems. Three of these layout problems will be discussed in this section.

All programs were run using identical data sets except for the value of the MUST input for ALDEP, because this value when changed can make some inprovement in ALDEP's layouts and the best value for it varies with the layout problem. As stated previously, all programs use the same random number generator to control the effect of random chance on the layouts. The best layouts produced by each program, the best layouts produced by hand methods (preassigned into the computer program for scoring verification and uniformity of appearance only) are all included at the end of this section.

The first layout problem is the one used in the example problem section: Recoil Starter Mfg Company. This is a hypothetical problem of course, but in constructing this problem and all the others, an attempt was made to make the data realistic so the layouts could be evaluated using common sense as well as the program's evaluation method. In this problem there are a total of 43 preferences to satisfy and 14

departments to place anywhere in the layout in order to satisfy them. Receiving and shipping departments were preassigned to make sure they would be placed at the outside edge of the building. ALRANDOM was run first to be used as a lower bound with which to compare the effectiveness of the other two programs. Table 2 shows the results of the various methods:

Table 2 Performance Comparison on Hypothetical Problem 1 *

Program	Best score	Worst score	Average	Attempts	Time
ALRANDOM	652	196	443	5000	7.3 min.
ALDEP	672	210	452	5000	3.7 min.
OPDEP	752	552	670	100	.8 min.
Hand methods	772	n/a	n/a	many	6 hrs.

* The theoretical perfect score of this job is 796.

A complete sample output of this problem is provided in Appendix A. From these results it appears that the only computerized method that generated a good enough layout to consider was OPDEP. With a reasonably small and uncomplicated layout problem such as this one, the noncomputer methods are superior.

The next layout problem to be discussed is an actual one

and concerns a medium sized agricultural products manufacturer located in a small town in Eastern Iowa. Data for the layout were collected in visits with the engineering department at the manufacturing facility. The actual material handling flow between workcenters was not directly available. The required data was gathered by study of the routing sheets and careful inspection of the material handling system. The company engineers felt the results would be most beneficial to the company if those departments which could not be moved, were constrained to their existing locations. Therefore, 11 of the total of 33 departments were preassigned for all the layouts. The facility is one floor, 300 feet by 225 feet; using a square size of 500 square feet, this equates to a layout matrix 10 squares by 13 squares. Using the program scoring criteria, the existing layout of this facility scores 415. The preference values used in this problem are A=24, E=7, I=3, O=1, U=0 and X=-9. Table 3 shows the results of the various methods.

Table 3 Performance Comparison on Actual Problem *

Program	Best score	Worst score	Average	Attempts	Time
ALRANDOM	566	94	312	5000	36.3 min.
ALDEP	582	72	310	5000	40.2 min.
OPDEP	792	370	571	500	6.0 min.
Hand methods	842	n/a	n/a	many	1 day

* The theoretical perfect score of this job is 940.

The only good design produced by a computer program was the one designed by OPDEP. The large increase in OPDEP's scores compared to the other two programs is partly due to the much larger number of random arrays that are possible with 33 departments vs 16 departments on the previous problem. The use of preassigned departments also tends to decrease the performance of the ALDEP program for the reasons outlined in Chapter II. The Engineering staff of this company who wished to remain anonymous, were favorably impressed by these results from OPDEP. They commented that they had been unaware that usable results such as these could be obtained from a computer program, and they expressed a desire to use the program themselves for this purpose.

The last layout problem to be presented here is another hypothetical factory called the Bradley Tractor company. This

problem was designed to demonstrate the performance of the four methods on a very large and complex layout design problem. This hypothetical company has 60 departments with a large product line, therefore the material handling flow between the departments is very high and most departments have a material handling flow to a large number of other departments. This problem has been artificially complicated in this manner to find what effect the complications will have on the performance of each of the design methods. The dimensions of the layout matrix are 8 squares by 10 squares on a single floor. The receiving departments and the shipping warehouse were preassigned to ensure they would all be placed on the periphery of the layout. The program is left with 57 departments to lay out. The preference values are A=40, E=32, I=20, O=14, U=6, and X=0. Table 4 shows the results of the various methods on this problem.

Table 4 Performance Comparison on Hypothetical Problem 2 *

Program	Best score	Worst score	Average	Attempts	Time
ALRANDOM	3770	1890	2640	500	19.7 min.
ALDEP	3326	1714	2604	500	21.9 min.
OPDEP	7104	4774	5939	150	3 min.
Hand methods	7156	n/a	n/a	many	3 days

* The theoretical perfect score of this job is 13,452.

ALRANDOM

TRIAL LAYOUT1521 SCORE= 676 EFFICIENCY= 84.9 %

TOP FLOOR

```
0  0
0  0
```

MAIN FLOOR

```
0  0  0  0  0  0  0
0  1  1  5 16 16  0
0  3  9  8 14 15  0
0  2  4 10 11 15  0
0  7 12  6 13  0  0
0  0  0  0  0  0  0
```

BASEMENT FLOOR

```
0  0
0  0
```

ALDEP

TRIAL LAYOUT +365 SCORE = 672

```
0  0
```

TOP FLOOR

```
0  0  0  0  0  0  0
0  1  1  5 16 16  0
0  3  3  8 11 15  0
0  4  9 10 13 15  0
0  2  6 12  0 14  0
0  0  0  0  0  0  0
```

MAIN FLOOR

```
0  0
```

TERRACE LEVEL

UNDER THE RULES FOR EVALUATION, THIS LAYOUT
SATISFIES ALL NECESSARY RELATIONSHIPS

TRIAL LAYOUT 59 RECOIL STARTER MF SCORE= 754 EFFICIENCY= 94.7 %
PERFECT SCORE = 796

TOP FLOOR

0	0
0	0

MAIN FLOOR

0	0	0	0	0	0	0
0	1	1	15	16	16	0
0	3	7	12	15	14	0
0	3	4	2	11	13	0
0	6	8	9	10	5	0
0	0	0	0	0	0	0

BASEMENT FLOOR

0	0
0	0

ACCORDING TO THE PREFERENCE TABLE INPUT FOR THIS
JOB AND NOT INCLUDING THOSE LESS THAN 1, THIS
LAYOUT DOES NOT SATISFY THESE PREFERENCES:

DEPT NBRS	TYPE PREF	DEPT NBRS	TYPE PREF	DEPT NBRS	TYPE PREF
105	101	0	101	105	0

** NOTE ** THE ABOVE LIST DOES NOT INCLUDE 4 PREFERENCES HALF SATISFIED
BECAUSE OF A SQUARE BETWEEN THEM, WHICH INCREASED THE SCORE OF THIS LAYOUT
BY 26. IT IS POSSIBLE TO SATISFY ALL PREFERENCES BY THESE CRITERIA AND NOT
GET A PERFECT SCORE

THIS LAYOUT SATISFIES 46 OF 48 PREFERENCES INPUT THAT ARE) OR = 1.

LEGEND OF DEPARTMENTS USED IN LAYOUT

1	RECEIVING RAW MAT	2	RECEIVING PURCH.	3	SHEAR
4	PUNCH PRESS	5	PLASTIC MOLDING	6	BRAKE PRESS
7	FORM PRESS-250T	8	ARC WELD	9	WELD CLEAN
10	SUBASSEMBLY	11	PAINTING	12	DEBURRING
13	DRILL PRESS	14	WIP (ASSY)	15	ASSEMBLY
16	SHIPPING/WAREHOUSE				

HAND METHODS (COMPUTER SCORED)

TRIAL LAYOUT 1 RECOIL STARTER MF SCORE= 772 EFFICIENCY= 97.0 %
PERFECT SCORE = 796

TOP FLOOR

```
0  0
0  0
```

MAIN FLOOR

```
0  0  0  0  0  0  0
0  1  1 15 16 16  0
0  3  5 12 15  2  0
0  3  7  8  9 14  0
0  4  6 10 11 13  0
0  0  0  0  0  0  0
```

BASEMENT FLOOR

```
0  0
0  0
```

ACCORDING TO THE PREFERENCES INPUT FOR THIS JOB,
THIS LAYOUT SATISFIES ALL PREFERENCES THAT ARE
GREATER THAN OR EQUAL TO 1.

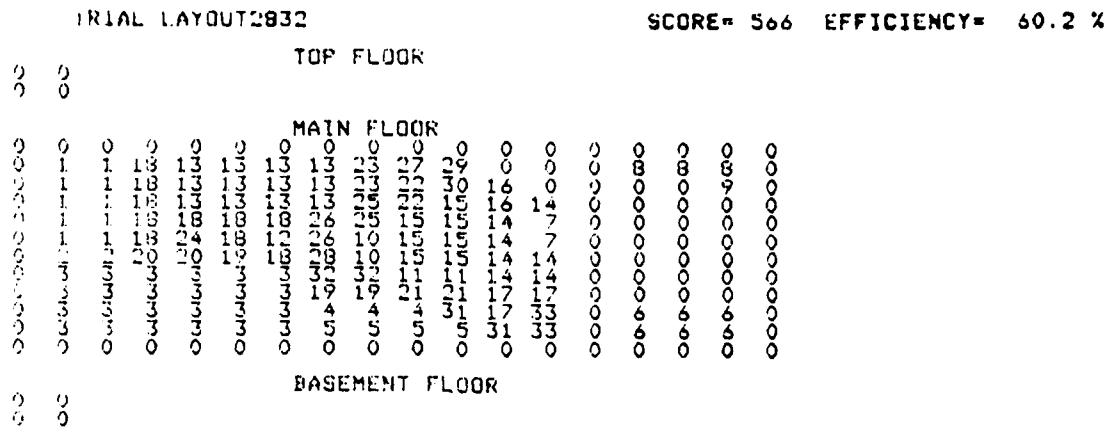
** NOTE ** THE ABOVE LIST DOES NOT INCLUDE 6 PREFERENCES HALF SATISFIED BECAUSE OF A SQUARE BETWEEN THEM, WHICH INCREASED THE SCORE OF THIS LAYOUT BY 18. IT IS POSSIBLE TO SATISFY ALL PREFERENCES BY THESE CRITERIA AND NOT GET A PERFECT SCORE

THIS LAYOUT SATISFIES 48 OF 48 PREFERENCES INPUT THAT ARE > OR = 1.

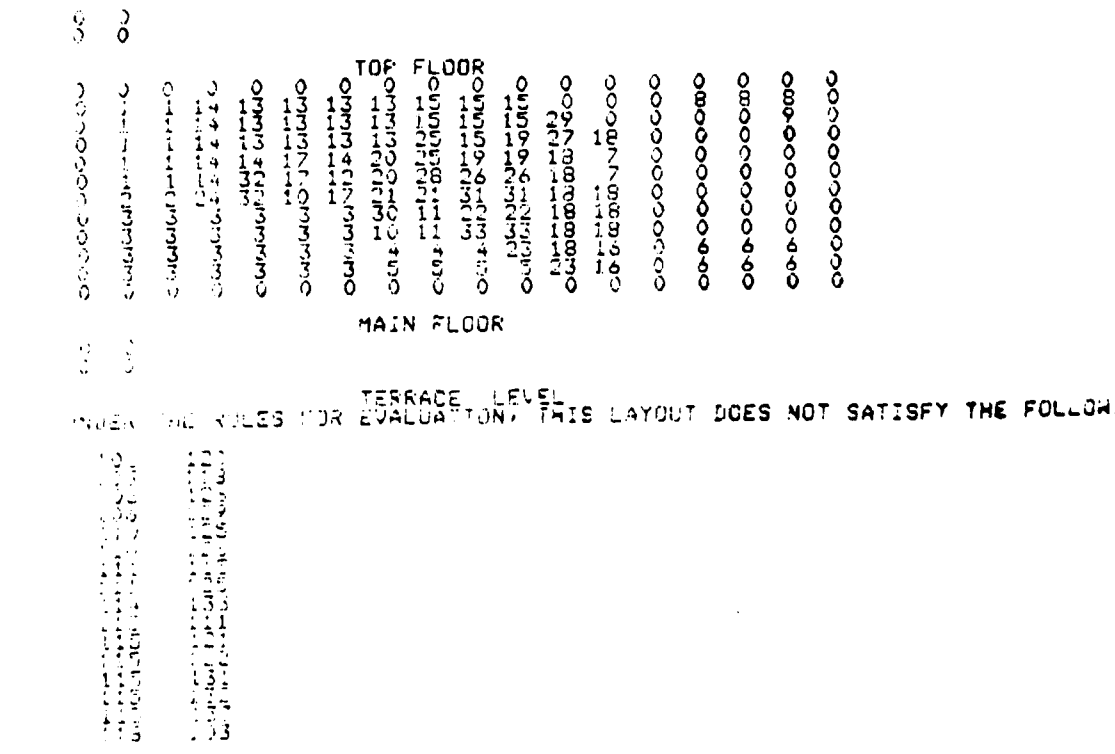
LEGEND OF DEPARTMENTS USED IN LAYOUT

1	RECEIVING RAW MAT	2	RECEIVING PURCH.	3	SHEAR
4	PUNCH PRESS	5	PLASTIC MOLDING	6	BRAKE PRESS
7	FORM PRESS-250T	8	ARC WELD	9	WELD CLEAN
10	SUBASSEMBLY	11	PAINTING	12	DEBURRING
13	DRILL PRESS	14	WIP (ASSY)	15	ASSEMBLY
16	SHIPPING/WAREHOUSE				

ALRANDOM



ALDEP



OPDEP

INITIAL LAYOUT 27 SCORE= 792 EFFICIENCY= 84.3 %
PERFECT SCORE = 940

TOP FLOOR

0	0
0	0

MAIN FLOOR

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	17	13	13	13	13	17	17	14	0	0	0	8	8	8	0
1	1	17	13	13	13	13	17	14	14	0	0	0	0	0	9	0
1	1	17	13	13	13	13	15	14	14	16	16	0	0	0	0	0
1	1	18	18	18	18	15	15	14	14	30	7	0	0	0	0	0
1	1	18	18	18	12	15	15	32	32	29	7	0	0	0	0	0
1	1	20	20	18	18	15	15	25	25	28	24	0	0	0	0	0
3	3	3	3	3	3	26	26	27	10	33	33	0	0	0	0	0
3	3	3	3	3	3	23	23	11	10	22	22	0	0	0	0	0
4	4	4	4	4	4	4	4	4	11	31	31	0	6	6	6	0
5	5	5	5	5	5	5	5	5	5	21	21	0	6	6	6	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BASEMENT FLOOR

0	0
0	0

ACCORDING TO THE PREFERENCE TABLE INPUT FOR THIS JOB AND NOT INCLUDING THOSE LESS THAN 1, THIS LAYOUT DOES NOT SATISFY THESE PREFERENCES:

DEPT	NBRS	TYPE PREF	DEPT	NBRS	TYPE PREF	DEPT	NBRS	TYPE PREF
110	106	E	126	106	U	129	106	U
125	113	E	133	114	E	121	115	I
122	115	I	131	115	E	133	115	U
102	118	E	125	118	U	125	118	E
129	118	U	115	121	I	133	121	E
115	122	I	118	122	U	126	124	E
114	125	E	118	125	E	106	126	U
134	126	E	118	128	U	102	128	U
115	131	E	114	133	E	115	133	U
121	133	E						

** NOTE ** THE ABOVE LIST DOES NOT INCLUDE 14 PREFERENCES HALF SATISFIED BECAUSE OF A SQUARE BETWEEN THEM, WHICH INCREASED THE SCORE OF THIS LAYOUT BY 14. IT IS POSSIBLE TO SATISFY ALL PREFERENCES BY THESE CRITERIA AND NOT GET A PERFECT SCORE

THIS LAYOUT SATISFIES 68 OF 96 PREFERENCES INPUT THAT ARE) OR = 1.

LEGEND OF DEPARTMENTS USED IN LAYOUT

1	FRONT OFFICE	2	PUR. PTS. RCVG.	3	SHIPPING/WHSE
4	STEEL STORAGE	5	TRUCK ACCESS	6	STEEL RCVG/FOAM
7	REST ROOMS EAST	8	HYDRANT MACHINING	9	COMPRESSOR ROOM
10	12 FT CINC. SHEAR	11	6 FT CINC. SHEAR	12	REST ROOMS WEST
13	COATING	14	ARC WELDING	15	SPOT WELDING
16	TOOL & DIE	17	SHOT BLASTING	18	ASSEMBLY
19	COATED PARTS STORAGE	20	PURCH. PARTS STG.	21	BLISS PRESS 166T
22	VERSON PRESS 75T	23	4 FT VERSON BRAKE	24	TUMBLING
25	BLISS PRESS 110T	26	BLISS PRESS 45T	27	HINGE ASSY
28	JOHNSON SMALL PRESS	29	CUT PIPE (621)	30	THREADING (642)
31	6 FT CHICAGO BRAKE	32	12 FT VERSON BRAKE	33	12 FT VERSON BRAKE

HAND METHODS (COMPUTER SCORED)

TRIAL LAYOUT 1 SCORE= 816 EFFICIENCY= 86.8 %
PERFECT SCORE = 940

TOP FLOOR

MAIN FLOOR

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	17	13	13	13	13	15	15	15	31	22	0	0	8	0
0	1	1	17	13	13	13	13	15	15	15	31	22	0	0	9	0
0	1	1	17	13	13	13	13	19	15	27	32	0	0	0	0	0
0	1	1	14	14	14	14	16	19	26	25	32	7	0	0	0	0
0	1	1	14	14	14	12	16	19	26	25	0	7	0	0	0	0
0	2	20	20	18	18	18	18	18	24	23	23	0	0	0	0	0
0	3	3	3	3	3	3	18	18	18	10	33	33	0	0	0	0
0	3	3	3	3	3	3	28	10	18	10	21	21	0	0	0	0
0	3	3	3	3	3	3	4	4	4	11	11	0	0	6	6	0
0	3	3	3	3	3	3	5	5	5	5	0	30	29	6	6	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BASEMENT FLOOR

According to the preference table input for this job and not including those less than 1, this layout does not satisfy these preferences:

DEPT	NBRS	TYPE PREF	DEPT	NBRS	TYPE PREF	DEPT	NBRS	TYPE PREF
110	106	E	126	106	0	115	111	0
122	111	1	125	111	0	131	111	0
125	113	E	132	113	E	133	114	E
111	115	0	110	115	0	121	115	I
133	115	0	106	118	E	115	118	0
122	118	0	115	121	I	131	121	E
111	122	I	110	122	0	133	122	0
111	125	0	113	125	E	106	126	0
111	131	0	121	131	E	113	132	E
114	133	E	115	133	0	122	133	0

** NOTE ** THE ABOVE LIST DOES NOT INCLUDE 12 PREFERENCES HALF SATISFIED BECAUSE OF A SQUARE BETWEEN THEM, WHICH INCREASED THE SCORE OF THIS LAYOUT BY 14. IT IS POSSIBLE TO SATISFY ALL PREFERENCES BY THESE CRITERIA AND NOT GET A PERFECT SCORE.

THIS LAYOUT SATISFIES 66 OF 96 PREFERENCES INPUT THAT ARE > OR = 1.

LEGEND OF DEPARTMENTS USED IN LAYOUT

1	FRONT OFFICE	2	PUR. PTS. RCVG.	3	SHIPPING/WHSE
4	STEEL STORAGE	5	TRUCK ACCESS	6	STEEL RCVG/FOAM
7	REST ROOMS EAST	8	HYDRANT MACHINING	9	COMPRESSOR ROOM
10	12 FT CINC. SHEAR	11	6 FT CINC. SHEAR	12	REST ROOMS WEST
13	COATING	14	ARC WELDING	15	SPOT WELDING
16	TOOL & DIE	17	SHOT BLASTING	18	ASSEMBLY
19	COATED PARTS STORAGE	20	PUNCH. PARTS STG.	21	BLISS PRESS 166T
22	VERSON PRESS 75T	23	4 FT VERSON BRAKE	24	TUMBLING
25	BLISS PRESS 110T	26	BLISS PRESS 45T	27	HINGE ASSY
28	JOHNSON SMALL PRESS	29	CUT PIPE (621)	30	THREADING (642)
31	6 FT CHICAGO BRAKE	32	12 FT VERSON BRAKE	33	12 FT VERSON BRAKE

ALRANDOM

TRIAL LAYOUT 186 BRADLEY MFG. SCORE=3770 EFFICIENCY= 27.3 %

TOP FLOOR

MAIN FLOOR

BASEMENT FLOOR

ALDEP

TRIAL LAYOUT 121R SCORE = 3326

TOP FLOOR

MAIN FLOOR

TERRACE LEVEL

UNDER THE RULES FOR EVALUATION, THIS LAYOUT

DOES NOT SATISFY THE FOLLOWING CLOSE RELATIONSHIPS

OPDEP

TRIAL LAYOUT 128 BRADLEY TRACTOR MFG. SCORE= 7104 EFFICIENCY= 52.8 %
PERFECT SCORE = 13452

TOP FLOOR

MAIN FLOOR

0	0	0	0	0	0	0	0	0	0	0
1	3	12	16	16	39	54	54	53	51	0
	3	4	15	38	36	52	52	46	46	0
	6	8	40	37	37	56	56	42	55	0
1	6	9	41	17	17	58	58	57	60	0
1	16	16	14	11	13	47	59	57	60	60	0
1	19	24	25	50	50	50	43	30	29	0	0
0	30	33	21	28	49	27	44	48	31	0	0
0	0	25	34	35	33	44	32	7	0	0
0	0	0	0	0	0	0	0	0	0	0

BASEMENT FLOOR

ACCORDING TO THE PREFERENCE TABLE INPUT FOR THIS JOB AND NOT INCLUDING THOSE LESS THAN 40, THIS LAYOUT DOES NOT SATISFY THESE PREFERENCES:

DEPT	NBRS	TYPE PREF	DEPT	NBRS	TYPE PREF	DEPT	NBRS	TYPE PREF
129	102	A	121	102	A	132	102	A
111	103	A	136	103	A	107	104	A
114	104	A	117	105	A	104	107	A
116	107	A	103	111	A	104	114	A
107	116	A	142	116	A	105	117	A
132	118	A	102	119	A	102	121	A
102	132	A	118	132	A	103	136	A
118	142	A	157	153	A	157	154	A
153	157	A	154	157	A			

** NOTE ** THE ABOVE LIST DOES NOT INCLUDE 98 PREFERENCES HALF SATISFIED BECAUSE OF A SQUARE BETWEEN THEM, WHICH INCREASED THE SCORE OF THIS LAYOUT BY1194. IT IS POSSIBLE TO SATISFY ALL PREFERENCES BY THESE CRITERIA AND GET A PERFECT SCORE

THIS LAYOUT SATISFIES 108 OF 134 PREFERENCES INPUT THAT ARE > OR = 40.

LEGEND OF DEPARTMENTS USED IN LAYOUT

1	RECV'G RAW MATERIALS	2	RAW MATERIAL WHSE.	3	SHEAR1
4	SHEAR2	5	SHEAR3	6	IRONWORKER1
7	IRONWORKER2	8	PUNCH PRESS 50T	9	PUNCH PRESS
10	PUNCH PRESS 25T	11	BRAKE 100T	12	BRAKE 50T
13	BRAKE 50T	14	FORM PRESS 200T	15	FORM PRESS
16	ARC WELDING	17	SPOT WELDING	18	MIG WELDING
19	LATHE1	20	LATHE2	21	LATHE3
22	TURRET LATHE1	23	VERT MILL1	24	VERT MILL2
25	HORIZ MILL1	26	HORIZ MILL2	27	TURRET LATHE2
28	BROACHING	29	CIRCLE SHEAR	30	SPIN MACH1
31	SPIN MACH2	32	BAND SAW	33	CUT OFF MACH
34	THREADER1	35	THREADER2	36	EDGE ROLLER
37	SUBASSEMBLY1	38	ROLLER1	39	ROLLER2
40	PAINT ASSY	41	SWEDGE PIN	42	DRILL PRESS1
43	FLAME CUTTER	44	DRILL PRESS2	45	PRESS 25T
46	PRESS 30T	47	DEBURRING1	48	DEBURRING2
49	SAND MOLD	50	CASTING	51	INJ. MOLDING
52	PAINTING	53	PLASTIC ASSY	54	SUBASSEMBLY2
55	PURCHASE PARTS REC	56	WIP ASSEMBLY	57	FINAL ASSEM
58	INSPECTION1	59	INSPECTION2	60	SHIPPING

HAND METHODS (COMPUTER SCORED)

TRIAL LAYOUT 2 BRADLEY TRACTOR MFG. SCORE= 7156 EFFICIENCY= 53.2 %
PERFECT SCORE = 13452

TOP FLOOR

0 0
0 0

MAIN FLOOR

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	13	16	16	43	45	46	56	0
0	0	0	3	4	15	38	36	39	51	56	0
0	0	0	0	8	40	17	32	54	54	53	55
0	0	1	0	0	9	41	17	37	59	57	57
0	0	0	0	14	11	11	43	58	57	60	60
0	0	0	0	4	25	50	50	43	30	29	0
0	0	0	0	21	28	49	27	44	48	31	0
0	0	0	0	15	34	35	33	44	32	7	0
0	0	0	0	0	0	0	0	0	0	0	0

BASEMENT FLOOR

0 0
0 0

ACCORDING TO THE PREFERENCE TABLE INPUT FOR THIS JOB AND NOT INCLUDING THOSE LESS THAN 40, THIS LAYOUT DOES NOT SATISFY THESE PREFERENCES:

DEPT NBRS	TYPE PREF	DEPT NBRS	TYPE PREF	DEPT NBRS	TYPE PREF
119 102	A	121 102	A	132 102	A
111 103	A	136 103	A	107 104	A
114 104	A	117 105	A	104 107	A
116 107	A	103 111	A	104 114	A
102 116	A	105 117	A	132 118	A
102 119	A	102 121	A	102 132	A
118 132	A	103 136	A		

** NOTE ** THE ABOVE LIST DOES NOT INCLUDE 96 PREFERENCES HALF SATISFIED BECAUSE OF A SQUARE BETWEEN THEM, WHICH INCREASED THE SCORE OF THIS LAYOUT BY1136. IT IS POSSIBLE TO SATISFY ALL PREFERENCES BY THESE CRITERIA AND GET A PERFECT SCORE

THIS LAYOUT SATISFIES 114 OF 134 PREFERENCES INPUT THAT ARE) OR = 40.

LEGEND OF DEPARTMENTS USED IN LAYOUT

1	REC'G RAW MATERIALS	2	RAW MATERIAL WHSE.	3	SHEAR1
4	SHEAR2	5	SHEAR3	6	IRONWORKER1
7	IRONWORKER2	8	PUNCH PRESS 50T	9	PUNCH PRESS
10	PUNCH PRESS 25T	11	BRAKE 100T	12	BRAKE 50T
13	BRAKE 50T	14	FORM PRESS 200T	15	FORM PRESS
16	ARC WELDING	17	SPOT WELDING	18	MIG WELDING
19	LATHE1	20	LATHE2	21	LATHE3
22	TURRET LATHE1	23	VERT MILL1	24	VERT MILL2
25	HORIZ MILL1	26	HORIZ MILL2	27	TURRET LATHE2
28	BROACHING	29	CIRCLE SHEAR	30	SPIN MACH1
31	SPIN MACH2	32	BAND SAW	33	CUT OFF MACH
34	THREADER1	35	THREADER2	36	EDGE ROLLER
37	SUBASSEMBLY1	38	ROLLER1	39	ROLLER2
40	PATCH ASSY	41	SWEDGE PIN	42	DRILL PRESS1
43	FLAME CUTTER	44	DRILL PRESS2	45	PRESS 25T
46	PRESS 30T	47	DEBURRING1	48	DEBURRING2
49	SAND MOLD	50	CASTING	51	INJ. MOLDING
52	PAINTING	53	PLASTIC ASSY	54	SUBASSEMBLY2
55	PURCHASE PARTS REC	56	WIP ASSEMBLY	57	FINAL ASSEM
58	INSPECTION1	59	INSPECTION2	60	SHIPPING

The methods of Chapter III were not used in these comparisons because they are intended to show the best results the programs can achieve without outside interaction by the user. If the methods of Chapter III are used to guide the program's designs, then layouts which exceed the best ones designed by hand methods can generally be achieved. In the Bradley Tractor problem scores in the 7300 range were obtained by this method. The next section discusses the relative performance of the programs.

Conclusions

The superiority of OPDEP over the other programs is even more apparent in the last problem than in the first two problems. The best scores generated by the ALDEP and ALRANDOM programs are only about 50% of the best score generated by OPDEP. The improvement in performance of OPDEP over the other programs seems to be proportional to the size and complexity of the problem involved. This is as one might expect in view of the logic which governs department placement in each of the programs. The superiority of ALDEP over ALRANDOM cannot be concluded from the results of these layout problems, nor from any of the other problems attempted by the author. It appears

that both programs produce approximately equal layout designs on any given problem and they are both more dependent on the random number generator than anything else. In all the layout problems attempted by the author, the noncomputer methods eventually enable one to design a layout that is better than any designed by a program without interaction by the user. However, using the techniques in Chapter III of this paper, even better designs can be made by the OPDEP program.

The OPDEP program is still limited somewhat in its performance by some of the inherent characteristics of the design algorithm. In this section some of these problems will be pointed out to the reader.

The OPDEP program is restricted to producing those designs which are possible by following the design pattern determined by the LAYOUT subroutine. This pattern can be varied by changing its parameters, but some designs are prevented from developing because of this pattern. It is possible that some of the eliminated designs would be very good ones, so it seems that a more variable pattern should be developed to allow more flexibility in the designs. Perhaps a design that starts in the middle and spirals outward to the matrix boundaries would be a better alternative.

Another fault of the program is a tendency for multi-square departments to be divided into several parts when a large number of preassigned departments are present.

Fortunately this does not happen often, but it is annoying when this problem ruins an otherwise good layout design. The changes required in the program to correct this problem would not be difficult to make.

An additional feature that would make the program more flexible, would be the added capability to include a preference for a department to be placed next to the periphery of the layout. This would be useful for ensuring that shipping and receiving departments or dock areas would be placed in a reasonable location on most layouts. It is now necessary to preassign a dummy department representing the walls of the facility and include a preference for this dummy department by the department of interest or to preassign the department to the periphery of the layout. The preassignment alternative places additional limitations on the layouts that are not always necessary and removes some of the flexibility from the layout process.

The program is also unable to look ahead as the design progresses to make decisions that will improve the design at a later point. This is the thing which usually allows the human mind to come up with better designs. This 'look ahead' concept is something that would be exceedingly difficult if not impossible, to implement in a computer program. This shortcoming is partially offset by the speed of the computer and the vast number of designs that can be made on each run.

It was not the author's objective in writing this paper to attempt to convince the reader of the superiority of the computer in plant layout design. The intent was rather to show that computer methods (OPDEP in particular), can be very useful in solving layout problems and they can eliminate most of the tedious work involved in attempting to find the optimum design combination. The final example in the previous section was a problem that most people would be reluctant to solve by conventional plant layout techniques. It is a problem comparable in some ways to a linear programming problem with 60 variables and 241 constraints (the number of distinct preferences in this problem's preference matrix). The marginal improvement in the score of the hand made layout, as compared to the one generated by OPDEP is so small that it may not have been worth the effort.

The designs generated by OPDEP are not to be viewed as final designs or optimal designs. They are rather to be viewed as reasonably good design alternatives that may be unrealistic in some respects, but are at least a good place to start applying analysis and reasoning towards some optimum solution.

REFERENCES

1. Muther, R. Systematic Layout Planning. Industrial Education Institute, Boston, Mass. 1961.

2. Hicks, P. E. and Cowan, T. E. "CRAFT-M for Layout Rearrangement". Industrial Engineering 8, No. 5:30. May, 1976.

3. Moore, J. M. "Computer Aided Facilities Design: An International Survey". International Journal of Production Research 12, No. 1:21. January, 1974.

4. Tompkins, J. A. and Reed, R. "Computerized Facilities Design". Technical Papers 24:75. 1973.

5. Lee, R. C. and Moore, J. M. "CORELAP". AIIE Proceedings 18:274. 1967.

6. Tompkins, J. A. and Moore, J. M. Computer Aided Layout: A User's Guide. American Institute of Industrial Engineers, Inc., Norcross, Georgia. 1978.

7. Moore, J. M. "Computer Program Evaluates Plant Layout Alternatives". Industrial Engineering 3, No. 8:19. August, 1971.

8. Diesenroth, M. P. and Apple, J. M. "A Computerized Plant Layout Analysis and Evaluation Technique". Technical Papers 23:121. 1972.

9. Seehof, J. M. and Evans, W. O. "Automated Layout Design Program". AIIE Proceedings 18:231. 1967.

10. DeGroot, M. H. Probability and Statistics. Addison-Wesley Publishing Co., Reading, Massachusetts. 1975.

11. Liewellyn, R. J. "Travel Charting-First Aid for Plant Layout". Journal of Industrial Engineering 10, No. 3:217. 1958.

12. Moore, J. M. Plant Layout and Design. The Macmillan Co., New York, New York. 1962.

APPENDIX A. SAMPLE OUTPUT FROM THE OPDEP PROGRAM

FACILITIES LAYOUT - DESIGN PROGRAM - OFDEP

STARTING NEW JOB, RUN CODE = RECOIL STARTER MF DATE DECEMBER 1, 1979

RANDOM NO. SEEDS ARE 1357 3579

SQUARE SIZE IS 250.0

NUMBER OF LAYOUTS REQUESTED IS 5

MIN ACCEPTABLE SCORE IS 730

VARIABLE FORMAT FOR THE LAYOUT MATRIX IS (02I3) (07I3) (02I3)

SWEEP LENGTH FOR TOP FLOOR 0

SWEEP LENGTH FOR MAIN FLOOR 2

SWEEP LENGTH FOR BASEMENT 0

INPUT MATRIX

```
101  S
102  E S
103  E E S
104  E E A S
105  O O O O S
106  U U O A U S
107  U U U I O U S
108  U U U O O U A S
109  U U U U U U U E S
110  U U U U U U U I O S
111  U U U U U U U U E U S
112  U U U U U U U U U O U S
113  U U U U U U U U U U O U S
114  U I U U U U U U U U I O U I S
115  U U U U U U U U U U U U U U A S
```

EXPLANATION OF PREFERENCES

A-REPRESENTS THE HIGHEST CATEGORY OF M/H ACTIVITY
E-REPRESENTS A LARGE AMOUNT OF M/H ACTIVITY
I-REPRESENTS A MEDIUM AMOUNT OF M/H ACTIVITY
O-REPRESENTS A SMALL AMOUNT OF M/H ACTIVITY
U-REPRESENTS NO MATERIAL HANDLING ACTIVITY
X-REPRESENTS NO M/H ACTIVITY

NUMERICAL VALUES OF PREFERENCES

A = 28

E = 24

I = 20

O = 7

U = 0

X = 0

SCORING OF LAYOUTS

IF TWO DEPARTMENTS ARE LAID OUT ADJACENT, EITHER DIAGONALLY OR PERPENDICULAR TO EACH OTHER, THE VALUE OF THEIR PREFERENCE IS ADDED TO THE LAYOUT SCORE. IF TWO DEPTS ARE SEPARATED BY ONE SQUARE DIAGONALLY OR PERPENDICULAR, HALF OF THEIR PREFERENCE IS ADDED TO THE SCORE. THOSE DEPTS SEPARATED BY MORE THAN ONE SQUARE HAVE NONE OF THEIR PREFERENCE ADDED TO THE SCORE.

DECODE MATRIX

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
101	0	024	0	7	0	0	0	0	0	0	0	0	0	0	0	0									
102	0	0	0	0	0	0	7	0	7	0	0	0	0	020	0										
103	24	0	028	0	720	0	0	0	0	0	0	0	0	0	0										
104	0	028	0	028	7	7	0	0	0	0	0	0	0	0	0										
105	7	0	0	0	0	0	0	0	0	0	0	7	0	0	0										
106	0	0	728	0	0	028	0	0	0	0	0	0	0	0	0										
107	0	020	7	0	0	0	0	020	0	7	0	0	0	0											
108	0	7	0	7	028	0	024	0	0	0	0	0	0	0	0										
109	0	0	0	0	0	024	0	024	0	0	0	0	0	0	0										
110	0	7	0	0	0	020	024	0	024	0	0	0	0	0	0										
111	0	0	0	0	0	0	0	024	0	0	720	0	0	0											
112	0	0	0	0	7	0	7	0	0	0	0	0	0	7	0										
113	0	0	0	0	0	0	0	0	0	0	7	0	020	0	0										
114	0	0	0	0	0	0	0	0	0	020	0	020	020	0	0										
115	020	0	0	0	0	0	0	0	0	0	0	7	020	0	028										
116	0	0	0	0	0	0	0	0	0	0	0	0	0	028	0										

TOTAL NUMBER OF SQUARES IN BLDG.= 20
TOP FLOOR SQUARES= 0
MAIN FLOOR SQS. = 20
BASEMENT SQUARES= 0

TOTAL SQUARES AVAILABLE TO LAYOUT= 16
TOP FLOOR SQUARES= 0
MAIN FLOOR SQS. = 16
BASEMENT SQUARES= 0

TOTAL PREASSIGNED SQS ALL FLOORS= 4
TOP FLOOR SQUARES= 0
MAIN FLOOR SQS. = 4
BASEMENT SQUARES= 0

TOTAL SQS REQD. FOR ALL DEPTS= 20

TOTAL BUFFER SQUARES AVAILABLE= 0

NO. DEPTS TO BE PLACED BY THE DESIGN PROGRAM= 14

NO. SQS TO BE PLACED BY THE DESIGN PROGRAM = 16

DEPARTMENT	REQUIRED AREA	NO. SQUARES	NAME
101	500.000	0	RECEIVING RAW MAT
102	250.000	1	RECEIVING PURCH.
103	500.000	2	SHEAR
104	250.000	1	PUNCH PRESS
105	250.000	1	PLASTIC MOLDING
106	250.000	1	BRAKE PRESS
107	250.000	1	FORM PRESS-250T
108	250.000	1	ARC WELD
109	250.000	1	WELD CLEAN
110	250.000	1	SUBASSEMBLY
111	250.000	1	PAINTING
112	250.000	1	DEBURRING
113	250.000	1	DRILL PRESS
114	250.000	1	WIP (ASSY)
115	500.000	4	ASSEMBLY
116	500.000	3	SHIPPING/WAREHOUSE

TRIAL LAYOUT 2 RECOIL STARTER MF SCORE= 730 EFFICIENCY= 91.7 %
PERFECT SCORE = 796

TOP FLOOR

MAIN FLOOR

0	0	0	0	0	0
1	3	2	16	16	0
3	3	15	15	13	0
4	3	7	14	12	0
5	5	10	11	5	0
0	0	0	0	0	0

BASEMENT FLOOR

ACCORDING TO THE PREFERENCE TABLE INPUT FOR THIS
JOB AND NOT INCLUDING THOSE LESS THAN 1, THIS
LAYOUT DOES NOT SATISFY THESE PREFERENCES:

DEPT NBRS		TYPE PREF	DEPT NBRS		TYPE PREF	DEPT NBRS		TYPE PREF
105	101	0	108	102	0	110	102	0
101	105	0	102	108	0	102	110	0

** NOTE ** THE ABOVE LIST DOES NOT INCLUDE 6 PREFERENCES HALF SATISFIED
BECAUSE OF A SQUARE BETWEEN THEM, WHICH INCREASED THE SCORE OF THIS LAYOUT
BY 18. IT IS POSSIBLE TO SATISFY ALL PREFERENCES BY THESE CRITERIA AND NOT
GET A PERFECT SCORE

THIS LAYOUT SATISFIES 42 OF 48 PREFERENCES INPUT THAT ARE > OR = 1.

LEGEND OF DEPARTMENTS USED IN LAYOUT

1	RECEIVING RAW MAT	2	RECEIVING PURCH.	3	SHEAR
4	PUNCH PRESS	5	PLASTIC MOLDING	6	BRAKE PRESS
7	FORM PRESS-250T	8	ARC WELD	9	WELD CLEAN
10	SUBASSEMBLY	11	PAINTING	12	DEBURRING
13	DRILL PRESS	14	WIP (ASSY)	15	ASSEMBLY
16	SHIPPING/WAREHOUSE				

END OF THIS JOB
TOTAL DESIGNS 5
TOTAL ACCEPTED ... 1

AVERAGE SCORE OF ALL LAYOUTS IS 670

MINIMUM SCORE IS 600 MAXIMUM SCORE IS 730

THE THEORETICAL PERFECT SCORE OF THIS JOB IS 796

SCORE PRINTOUT SUPPRESSED

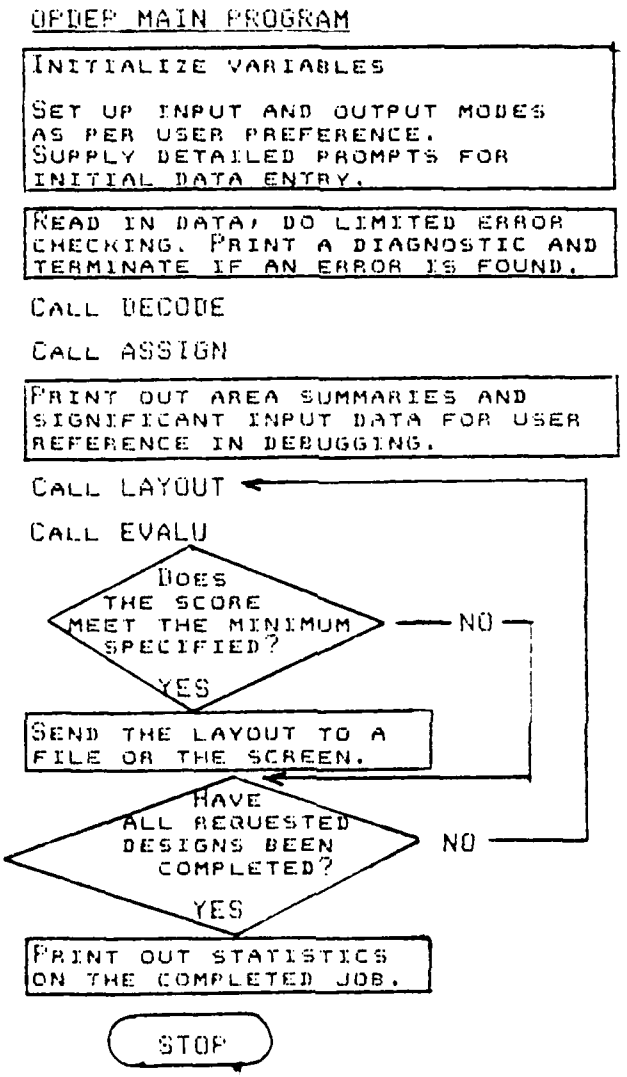
APPENDIX B. PROGRAM TO TEST RANDOM NUMBER GENERATOR

```
30-JAN-1980 14:05:05    VAX-11 FORTRAN IV-PLUS V1.3-22
                        UNITST.FOR.1

C  THE PURPOSE OF THIS PROGRAM IS TO TEST THE PERIOD OF
C  THE VAX PORTLIBRARY FUNCTION UNIFORMLY DISTRIBUTED
C  RANDOM NUMBER GENERATOR-UNI, WHEN USED TO GENERATE
C  AN ARRAY OF RANDOMLY ORDERED NUMBERS. THIS ARRAY IS
C  LIMITED TO THE RANGE 1 TO THE ARRAY SIZE WITH NO
C  DUPLICATE NUMBERS ALLOWED IN THE ARRAY.
C
C  FIRST SET UP THE ARRAY SIZE AND SEEDS.
      DIMENSION NRANDM(63),NTEST(63)
      PRINT *,'ENTER RANDOM GENERATOR SEEDS'
      READ *, NMOD, NMOD1
      CALL RANSET (NMOD,NMOD1)
      PRINT *,'ENTER NTIMES AND NOD'
      READ *, NTIMES,NOD
      PRINT *,'ENTER MAXIMUM CYCLES TO TRY'
      READ *,MAX
      XNOD=NOD-1
      NSUM=0
      NUM=0
C
C  NOW GENERATE THE ARRAY FOR WHICH TO FIND A DUPLICATE
C  OF.
C
      DO 1000 I=1,NTIMES
        KNT=0
        NO=1
        DO 2099 J=1,NOD
          NRANDM(J)=0
 2099   CONTINUE
  210   YFL= UNI(0)
        N=IFIX(YFL*XNOD+.5) +1
  212   DO 2126 J=1,NO
          IF(NRANDM(J)-N) 2121,210,2121
 2121     IF(NRANDM(J)) 2126,2127,2126
 2126   CONTINUE
 2127   NRANDM(NO)=N
        NO=NO+1
        IF(NO-NOD) 210,210,2128
C
C  NOW PLACE THIS OBJECT ARRAY IN AN ARRAY CALLED NTEST.
C
 2128   DO 50 K=1,NOD
          NTEST(K)=NRANDM(K)
   50   CONTINUE
C
C  NOW BEGIN GENERATING A SERIES OF ARRAYS TO COMPARE
C  WITH THE ARRAY IN NTEST.
C
  500   NO=1
        DO 3099 JJ=1,NOD
          NRANDM(JJ)=0
 3099   CONTINUE
  310   YFL=UNI(0)
        N=IFIX(YFL*XNOD+.5)+1
  312   DO 3126 JJ=1,NO
          IF(NRANDM(JJ)-N) 3121,310,3121
```

```
30-JAN-1980 14:05:05     VAX-11 FORTRAN IV-PLUS V1.3-22
                         UNITST.FOR.1

 3121      IF(NRANDM(JJ))3126,3127,3126
 3126    CONTINUE
 3127    NRANDM(NO)=N
         NO=NO+1
         IF(NO-NOD)310,310,3128
 3128    KNT=KNT+1
         IF(KNT.GT.MAX) GO TO 9999
C
C  NOW COMPARE EACH GENERATED ARRAY WITH NTEST TO SEE
C  IF A DUPLICATE HAS BEEN FOUND.
C
         DO 200 IJ=1,NOD
           IF(NRANDM(IJ).EQ.NTEST(IJ)) THEN
             IF(IJ.EQ.NOD) THEN

               IF A MATCH IS FOUND PRINT A MESSAGE
               THEN GENERATE A NEW NTEST ARRAY.

               PRINT *,'A DUPLICATE NRANDM ARRAY HAS'
               PRINT *,'BEEN FOUND IN',KNT,'CYCLES.'
               NSUM=NSUM+KNT
               GO TO 1000
             END IF
           ELSE
             GO TO 500
           END IF
  200    CONTINUE
         GO TO 500
 1000    CONTINUE
 9999    PRINT *,'KNT=',KNT
         IF(KNT.GT.MAX) THEN
            NSUM=NSUM+KNT
            NCYCAV=NSUM/I-1
         ELSE
            NCYCAV=NSUM/NTIMES
         END IF
         PRINT *,'AVERAGE # CYCLES BETWEEN DUPLICATE ARRAYS'
         PRINT *,'IS',NCYCAV
         PRINT *,'FOR AN ARRAY SIZE OF',NOD
         STOP
         END
```

APPENDIX C. FLOW CHART OF OPDEP

FLOW CHART FOR OPDEP



SUBROUTINE DECODE

READ IN THE USER SUPPLIED VALUES CORRESPONDING TO A,E,I,O,U,X.

↓

CONVERT THE A,E,I,O,U,X MATRIX TO A CORRESPONDING MATRIX OF INTEGERS.

↓

RETURN

SUBROUTINE ASSIGN

READ IN THE PREASSIGNED DEPARTMENTS AND TABULATE THEM INTO A MATRIX.

↓

COMPUTE PREASSIGNED AREA TOTALS ON ALL THE FLOORS.

↓

RETURN

SUBROUTINE EVALU

EVALUATE THE SUCCESS OF THE LAYOUT AND SELECT SUBROUTINES IN DESIGNING A LAYOUT WHICH MEETS THE CONDITIONS OF THE PREFERENCE MATRIX.

↓

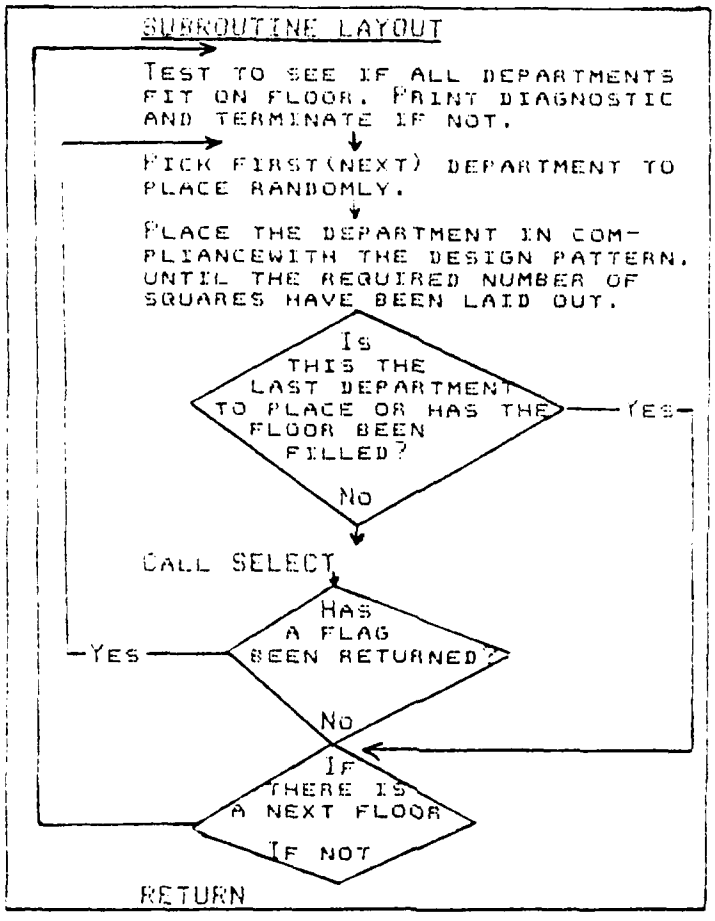
EACH PREFERENCE FULLY SATISFIED RECEIVES FULL VALUE, EACH PREFERENCE HALF SATISFIED RECEIVES HALF VALUE.

↓

CALCULATE THE INCREASE IN SCORE CAUSED BY CONSIDERING DEPARTMENTS SEPARATED BY ONE SQUARE AS WELL AS THOSE THAT ARE IMMEDIATELY ADJACENT TO EACHOTHER.

↓

RETURN



SUBROUTINE SELECT

FIND ALL THE DEPARTMENTS WHICH SURROUND THE NEXT SQUARE TO LAYOUT.

↓

SEARCH THE PREFERENCE MATRIX FOR ALL ALTERNATIVE DEPARTMENTS TO CHOOSE FROM.

↓

EVALUATE ALL ALTERNATIVES AND CHOOSE THE DEPARTMENT WHICH WILL MAXIMIZE THE THE LAYOUT SCORE AT THIS POINT.

↓

RETURN THIS DEPARTMENT # TO LAYOUT.

↓

IF NO DEPARTMENT CAN BE FOUND IN THE PREFERENCE MATRIX WHICH HAS A PREFERENCE FOR ANY DEPARTMENT SURROUNDING THE NEXT SQUARE TO LAYOUT, RETURN A FLAG TO INDICATE THIS.

↓

RETURN


ND
ATE